

Bazy danych

a. Projektowanie relacyjnej bazy danych				
b. 2	c. 3	d. 4	e. 5	f. 6
Uczeń: Podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych. Podaje przykłady programów do tworzenia baz danych. Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja). Potrafi utworzyć prostą kwerendę, jeden formularz i raport.	Uczeń: Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych. Określa podstawowe pojęcia (<i>rekord, pole, typ pola, relacja, klucz podstawowy</i>). Tworzy bazę danych składającą się z dwóch tabel, w każdej po kilka pól różnych typów. Projektuje formularze i raporty. Tworzy proste kwerendy wybierające. Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów. Zna zasady przygotowania korespondencji seryjnej.	Uczeń: Projektuje relacyjną bazę danych (na zadany temat) składającą się z trzech tabel połączonych relacją. Omawia typy relacji w bazie danych. Zna zasady definiowania kluczy podstawowych. Projektuje formularze i raporty według wskazówek nauczyciela. Potrafi utworzyć formularz z podformularzem. Umieszcza przyciski nawigacyjne. Tworzy kwerendy wybierające. Importuje dane z tabel arkusza kalkulacyjnego i dokumentu tekstowego do tabel bazy danych. Eksportuje dane z tabel bazy danych do tabel arkusza kalkulacyjnego i do dokumentu tekstowego.	Uczeń: Potrafi wytłumaczyć pojęcie relacji. Projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji). Zna kilka rodzajów formularzy i raportów. Umie zaprojektować samodzielnie wygląd formularza i raportu. Na formularzach umieszcza pola kombi, ogranicza wartości, wstawia (gdy jest taka potrzeba) bieżącą datę, umieszcza przyciski poleceń. Stosuje funkcje standardowe w kwerendach i standardowe operatory w kryteriach wyszukiwania. Korzysta z parametrów w kwerendzie.	Uczeń: Zna dokładnie wybrany program do projektowania baz danych. Potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu. Korzysta z dodatkowej, fachowej literatury.

Tworzenie kwerend z wykorzystaniem języka SQL				
2	3	4	5	6
Uczeń: Wyszukuje informacje w bazie, korzystając wyłącznie z gotowych kwerend i narzędzi wbudowanych do programu.	Uczeń: Tworzy samodzielnie kwerendy (proste i złożone), korzystając z wbudowanych do programu narzędzi. Wie, co to jest język SQL. Potrafi przeanalizować przykład zapytania utworzonego w języku SQL.	Uczeń: Zna zasady wyszukiwania informacji w bazie z wykorzystaniem języka zapytań. Zna składnię i działanie podstawowych instrukcji. Potrafi zapisać prostą kwerendę, korzystając z języka zapytań.	Uczeń: Potrafi zapisać złożone kwerendy, korzystając z wybranej instrukcji. Stosuje instrukcję <code>SELECT</code> i jej główne klauzule, by wybrać kolumny z tabel bazy danych. Wykorzystuje klauzulę <code>JOIN</code> do łączenia informacji z wielu tabel i kwerend oraz przedstawiania wyników jako jednego logicznego połączenia rekordów. Stosuje instrukcje <code>INSERT</code> do dopisywania rekordów i <code>UPDATE</code> do modyfikowania rekordów w bazie. Usuwa rekordy, korzystając z instrukcji <code>DELETE</code> .	Uczeń: Opierając się na profesjonalnej literaturze, potrafi samodzielnie zapisywać złożone kwerendy z wykorzystaniem języka zapytań SQL.

Realizacja projektu programistycznego				
2	3	4	5	6
Uczeń: Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje. Bierze udział w testowaniu projektu.	Uczeń: Zna wszystkie etapy projektowania systemów informatycznych. Uczestniczy czynnie w poszczególnych etapach projektu, wykonując zleczone zadania szczegółowe. Planuje temat projektu.	Uczeń: Wie, co to jest system informatyczny. Potrafi omówić zakres prac na każdym etapie. Realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania. Uczestniczy czynnie w analizie systemu informacyjnego, przygotowuje dokumentację. Przygotowuje założenia w postaci dokumentów edytora tekstu, korzystając z szablonów. Współpracuje przy projektowaniu tabel, formularzy i raportów.	Uczeń: Realizuje projekt zgodnie z zamierzoną organizacją pracy zespołowej i wytyczonymi wcześniej etapami projektowania. Wykonuje trudniejsze prace związane z projektowaniem systemu. Przeprowadza analizę systemu informacyjnego. Projektuje złożone kwerendy, formularze, raporty. Uczestniczy we wdrażaniu systemu informatycznego.	Uczeń: Potrafi pełnić funkcję koordynatora projektu. Przydziela zadania szczegółowe, dba o ich prawidłowe wykonanie, nadzoruje pracę innych, dba o dobrą atmosferę w grupie.

g.

Algorytmika i programowanie – problemy zaawansowane

h. Wprowadzanie danych i wyprowadzanie wyników				
i. 2	j. 3	k. 4	l. 5	m. 6
Uczeń: Zna i stosuje podstawowe sposoby wprowadzania danych i wyprowadzania wyników w wybranym języku programowania.	Uczeń: Potrafi poprawnie zadeklarować dane potrzebne do rozwiązania zadania. Wie, co określa typ danych i format danych. Potrafi odpowiednio sformatować wyprowadzane wyniki.	Uczeń: Tworzy własne typy danych (typ okrojony, typ wycieniowy) i stałe. Zna procedury i funkcje umożliwiające generowanie danych losowych. Stosuje w programach generowanie losowych danych.	Uczeń: Poprawnie formatuje wyprowadzane wyniki. Omawia stos jako przykład struktury danych. Wykonuje operacje na stosie. Implementuje stos z wykorzystaniem tablicy. Zna i omawia zastosowanie stosu (odwrotną notację polską – ONP).	Uczeń: Dobiera sposób wprowadzania danych do rozwiązywanego zadania. Potrafi sprawdzić poprawność danych wprowadzanych do programu. Definiuje własne typy danych, potrzebne do rozwiązania danego zadania. Korzystając z możliwości generowania danych losowych, tworzy własne gry komputerowe.

Wybrane typy i struktury danych, w tym dynamiczne				
2	3	4	5	6
Uczeń: Zna i omawia strukturę typów danych w wybranym języku programowania. Wie, jakiego typu dane zapisuje się w rekordzie. Wie, czym charakteryzują się dane typu tablicowego.	Uczeń: Tworzy prosty program, w którym deklaruje dane typu rekordowego (strukturalnego). Wczytuje dane do rekordu. Potrafi omówić ogólnie zasady przetwarzania plików w wybranym języku programowania.	Uczeń: Potrafi zastosować rekordowy typ danych do przetwarzania danych różnego rodzaju. Zna zasady przetwarzania plików w wybranym języku programowania (w językach programowania). Stosuje odpowiednie procedury i funkcje. Deklaruje zmienne typu plikowego. Korzystając z przykładów, odczytuje dane z pliku i zapisuje dane do pliku.	Uczeń: Wykonuje operacje na plikach w wybranym języku programowania (w językach programowania). Przetwarza pliki tekstowe. Tworzy własne programy, w których wykorzystuje przetwarzanie plików. Zna dynamiczne struktury danych i typ wskaźnikowy danych. Deklaruje zmienne typu wskaźnikowego. Stosuje w programach zmienne wskaźnikowe. Tworzy zmienne	Uczeń: Stosuje w programach wybrane struktury dynamiczne. Tworzy listę jednokierunkową. Tworzy i przegląda binarne drzewo poszukiwań. Pisze program wyszukujący wartości w binarnym drzewie uporządkowanym. Korzystając z dodatkowej literatury, zapoznaje się z innymi strukturami dynamicznymi, np. z listą dwukierunkową. Tworzy programy z zastosowaniem struktur

			dynamiczne. Zna wybrane struktury dynamiczne. Analizuje gotowe programy, w których zastosowano listę jednokierunkową i binarne drzewo poszukiwań; uruchamia je i testuje dla wybranych danych.	dynamicznych. Rozwiązuje zadania z matury i olimpiady informatycznej i bierze w niej udział.
--	--	--	--	---

Programowanie modułowe i obiektowe

2	3	4	5	6
Uczeń: Wymienia modele programowania. Omawia model programowania strukturalnego. Wie, jakie są korzyści z definiowania procedur i funkcji. Potrafi zdefiniować procedury i funkcje w wybranym języku programowania. Wie, co to jest moduł.	Uczeń: Omawia szczegółowo modele programowania. Zna zasady programowania modułowego. Analizując przykładowe programy, tworzy własne moduły.	Uczeń: Wie, na czym polega programowanie obiektowe. Zna podstawowe pojęcia programowania obiektowego: <i>klasa, obiekt, pola, metody</i> .	Uczeń: Rozumie i potrafi zastosować typ obiektowy. Definiuje klasy. Deklaruje pola prywatne. Omawia cechy programowania obiektowego: dziedziczenie i polimorfizm. Wie, czym są metody wirtualne. Poznaje zastosowanie tych cech, analizując gotowe programy. Modyfikuje programy według wskazówek nauczyciela.	Uczeń: Tworzy własne programy, stosując poznane zasady programowania modułowego i obiektowego. Korzysta z fachowej literatury. Wie, na czym polega programowanie zdarzeniowe. Rozwiązuje zadania z olimpiady informatycznej i bierze w niej udział.

Wybrane algorytmy				
2	3	4	5	6
Uczeń: Zna i potrafi omówić wybrane algorytmy sortowania. Analizuje gotowe listy kroków wybranych algorytmów sortowania, wykonuje algorytmy dla wybranych danych. Potrafi omówić algorytm rozkładu liczby na czynniki pierwsze. Zna łamigłówkę Wież Hanoi. Wykonuje praktyczne ćwiczenie, odpowiednio przekłada trzy krążki.	Uczeń: Zna jeden z wybranych algorytmów sortowania: np. przez wstawianie. Sprawdza liczbę porównań elementów w tym algorytmie. Zna algorytmy zamiany liczb między dowolnymi systemami pozycyjnymi: obliczania wartości dziesiętnej liczby, zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie. Wie, jak sprawdzić, czy dana liczba jest liczbą doskonałą. Zna reprezentację danych numerycznych w komputerze: reprezentację binarną liczb ujemnych, reprezentację stałopozycyjną liczb, reprezentację zmiennopozycyjną liczb.	Uczeń: Zna algorytmy sortowania: np. przez wstawianie, przez scalanie, metodą szybką. Sprawdza liczbę porównań elementów w przypadku każdego z algorytmów. Zna algorytm przeszukiwania binarnego. Potrafi utworzyć listę kroków tego algorytmu. Zna przykładowe algorytmy na liczbach naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia wybrany algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego. Zna sposób zapisu liczby całkowitej i rzeczywistej (zmiennoprzecinkowej).	Uczeń: Omawia i stosuje w zadaniach algorytmy na liczbach naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia algorytm szybkiego podnoszenia do potęgi i algorytmy badające własności geometryczne (np. przynależność punktu do odcinka, badanie położenia punktu względem prostej). Omawia wybrany algorytm numeryczny, np. wyznaczenie miejsca zerowego funkcji, obliczanie wartości pierwiastka kwadratowego, obliczanie pola obszaru ograniczonego. Zna właściwości arytmetyki komputerowej. Na konkretnych przykładach potrafi sprawdzić, jak zmienia się wartość błędu względnego. Tworzy program komputerowy, stosując wybrany algorytm. Wykonuje projekt programistyczny, stosując zasady pracy zespołowej.	Uczeń: Zapisuje wybrane algorytmy sortowania (np. przez wstawianie, przez scalanie, metodą szybką) w postaci programu komputerowego. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Tworzy programy komputerowe, stosując wybrane algorytmy. Rozwiązuje zadania z matury i olimpiady informatycznej.

Aplikacje bazodanowe dostępne za pośrednictwem Sieci

n. Aplikacje bazodanowe dostępne za pośrednictwem Sieci				
o. 2	p. 3	q. 4	r. 5	s. 6
Uczeń: Wymienia przykładowe programy do projektowania i tworzenia stron internetowych. Potrafi wymienić podstawowe elementy, z których składa się strona WWW. W stopniu podstawowym posługuje się wybranym programem do tworzenia stron. Zna podstawowe znaczniki języka HTML. Tworzy nieskomplikowaną stronę internetową. Wstawia tytuł, formatuje tekst, umieszcza obraz. Wie, na czym polega technologia dynamicznego generowania stron internetowych.	Uczeń: Potrafi omówić strukturę pliku w języku HTML. Zna podstawy języka znaczników HTML i potrafi wykonać prostą stronę na zadany przez nauczyciela temat. Projektuje wygląd strony. Planuje jej zawartość (teksty, rysunki, dźwięki, animacje) i umieszcza na niej ww. elementy. Zna zasady dynamicznego przetwarzania stron internetowych. Potrafi napisać proste skrypty w języku PHP.	Uczeń: Potrafi samodzielnie zaprojektować wygląd strony. Zna reguły poprawnego projektowania układu strony, m.in. dba o jej czytelność i przejrzystość, o poprawność redakcyjną, i merytoryczną oraz prawną umieszczanych na niej tekstów i materiałów. Zna zaawansowane możliwości języka HTML: tabele, ramki, style. Zna sposoby publikowania stron w Internecie. Tworzy skrypty w języku PHP. Wyświetla dane instrukcją echo. Stosuje kodowanie UTF-8. Stosuje zmienne i operatory. Wie, jak utworzyć witrynę internetową opartą na bazie danych, m.in. w jaki sposób utworzyć prostą księgę gości w MySQL.	Uczeń: Potrafi samodzielnie wykorzystać poznane funkcje języka HTML do udoskonalenia własnych stron internetowych Włącza licznik odwiedzin na stronie. Dodaje inne typowe elementy: forum, księgę gości. Zna podstawy języka skryptowego PHP (lub JavaScript). Używa go dla osiągnięcia nieskomplikowanych efektów wizualnych na stronie. Tworzy skrypty przesyłające dane za pomocą formularzy HTML. Pisze kod wyświetlający prosty formularz i odbierający dane z formularza. Tworzy witrynę internetową opartą na bazie danych. Tworzy konta użytkownika i bazy danych na serwerze MySQL. Wykonuje zapytania do bazy danych z poziomu PHP. Tworzy prostą księgę gości: tworzy tabelę na wpisy z księgi gości, dodaje wpisy do księgi gości za pomocą instrukcji INSERT, tworzy formularz dodający wpisy do bazy danych, odczytuje dane z bazy za	Uczeń: Potrafi samodzielnie zapoznać się z nowym programem do tworzenia stron internetowych. Potrafi posługiwać się językiem skryptowym PHP (lub JavaScript) do tworzenia stron dynamicznych. Tworzy samodzielnie rozbudowaną witrynę internetową opartą na bazach danych.

			pomocą instrukcji SELECT.	
--	--	--	---------------------------	--